

Breakout Group #1: Functional Requirements (User Actions, Scenarios, and Rules)

User Actions are the basis for communicating requirements to a development team, as well as managing scope on a project.

Remember user actions answer the question Who Does What?

#1) Looking at the Email app on an iPhone or iPod touch, identify the User Actions that Apple implemented behavioral responses to, and list them in the table below or on poster board.

USER ACTIONS	
User views a list of photo albums.	
User selects a photo album to view.	

Scope is the set of user actions to be included in a given version of your app.

#2) Discuss the scope of each version of the Mail app as iOS has evolved over the years. Did you know the user action named 'User Searches Inbox' was not part of the Mail app in iOS 1.0? Discuss with the group how you would represent this evolution of the Mail app in your scope document.

Scenarios are user actions, plus detailed responses (behaviors) that a developer builds an app to perform.

#3) On a new piece of paper, detail the steps a user would follow in a scenario titled “User Checks His/Her Email”. Begin your scenario with “User launches the Mail app.” Describe in words what he/she will see, and what options (buttons) he/she is presented with on the screen (e.g. “User optionally clicks the Refresh Mail button.”)

Rules are conditional (branching) logic. If a certain condition is met, then something happens.

#4) Pretend you work for Apple. You want to make the Mail app a universal app so it runs on the iPhone and iPad. Discuss the differences in behavior you want to see on these devices, and then enhance your scenario details to include rules (branching logic) that describe these differences.

Breakout Group #2: Non-Functional Requirements & Project Priorities

Non-functional requirements are requirements that specify criteria used to judge the operation of a system, rather than specific behaviors.

#1) Look up Non-Functional Requirements on Wikipedia, where you will find a list of 47 examples. As a group, discuss the meaning of each of the examples, and decide which non-functional requirements are most applicable and important to iOS apps. List these non-functional requirements in the table below or on poster board.

#2) Price, quality, and speed are often priorities that people consider, but as you can see there are plenty more priorities to give thought to. If you don't already have low price, high quality work products, and speed of development (a.k.a. time to market) on your list of non-functional requirements consider adding them. Next, rank these Non-Functional Requirements in terms of their priority.

#3) Discuss apps and experiences that come to your mind as good and not-so-good examples of each non-functional requirement listed.

#4 Think and talk about reasonable requirements for a children's app, related to each of the non-functional requirements you've listed. Try to be specific, and articulate measurable results where possible (e.g. load time for each screen should be <3 seconds).

NON-FUNCTIONAL REQUIREMENT NAME	IMPORTANCE (1=LOW, 5=HIGH)	REASONABLE REQUIREMENT FOR A CHILDREN'S APP
Load Time	1	Upon launch the app should be available within 3 seconds to use.
Availability	3	Server based high scores list should be available 99.5% of the time.

Breakout Group #3: Prototyping Screen Mock-Ups (A.K.A Wire Frames)

A picture is worth a 1000 words. Discuss among your group your experience with or ideas about prototyping iOS apps.

What has worked for you in the past? What didn't work?

Make a list of ideas and/or tools you've used to build prototypes of a new app.

Google the term "Prototyping iPad apps" and see what you can find.

TOOL
Good ole' pen and paper
Adobe photoshop templates
Stencils

Breakout Group #4: Coding – Learning Objective C and Xcode

#1) Discuss your programming experience with your group, whether you've done any programming before, what languages you know. What resources worked for you for learning programming languages in the past?

#2 Make a list of resources available for learning Objective C and Xcode.

Start with the iTunes Store. Go to iTunes U and search on iPhone Programming. Make sure everyone in the group sees the results of this search. Also make sure to click See All since only the first 8 options are shown.

Next search Amazon.

RESOURCES FOR LEARNING OBJECTIVE C AND XCODE
iTunesU – FREE online video courses including the Stanford series. Paul Hegarty's Fall 2010 is a great option.
Book: iPhone Programming: The Big Nerd Ranch Guide is a great option.

#3 Following developers on twitter is a great way to learn what's happening in the world of Objective C and Xcode. Take a look at this list online of top developers to follow, and identify those you recognize or are interested in. <http://twitter.com/#!/ejknapp/top-devs-must-read/members>

Follow individuals in that list or follow the whole list. Note that although it is not obvious in many of the profiles, many individuals in the above list work for Apple.

TWITTER ACCOUNT	DESCRIPTION
@jeff_lamarche	Programmer & author of many books focusing on iPhone & Mac programming
@Jury	Mac and iPhone Developer Tools and Performance Evangelist at Apple.
@AaronHillegass	Author of Cocoa Programming for Mac OS X and iPhone Programming: the Big Nerd Ranch Guide. Founder of Big Nerd Ranch, Inc.

Author: Steve Glinberg, Founder, 123 Apps
Email: steve@123ColorApp.com
Twitter: @123ColorAppDev

Breakout Group #5: Outsourcing Development

#1) Discuss your experience with outsourcing software development among the group and answer the following questions:

- What were your experiences?
- Did you outsource to a local person or company, or to somebody working remotely?
- Did you outsource to somebody outside of the US?
- What went right?
- What went wrong?
- What did you learn?
- What are some of the challenges of working with someone not located near you, and what can you do to minimize the impact of these challenges?

#2) Assemble a list of resources for finding individual developers and/or iOS development consulting companies. Use Google to help you.

RESOURCES FOR FINDING iOS DEVELOPERS
Freelancer.com
ODesk.com